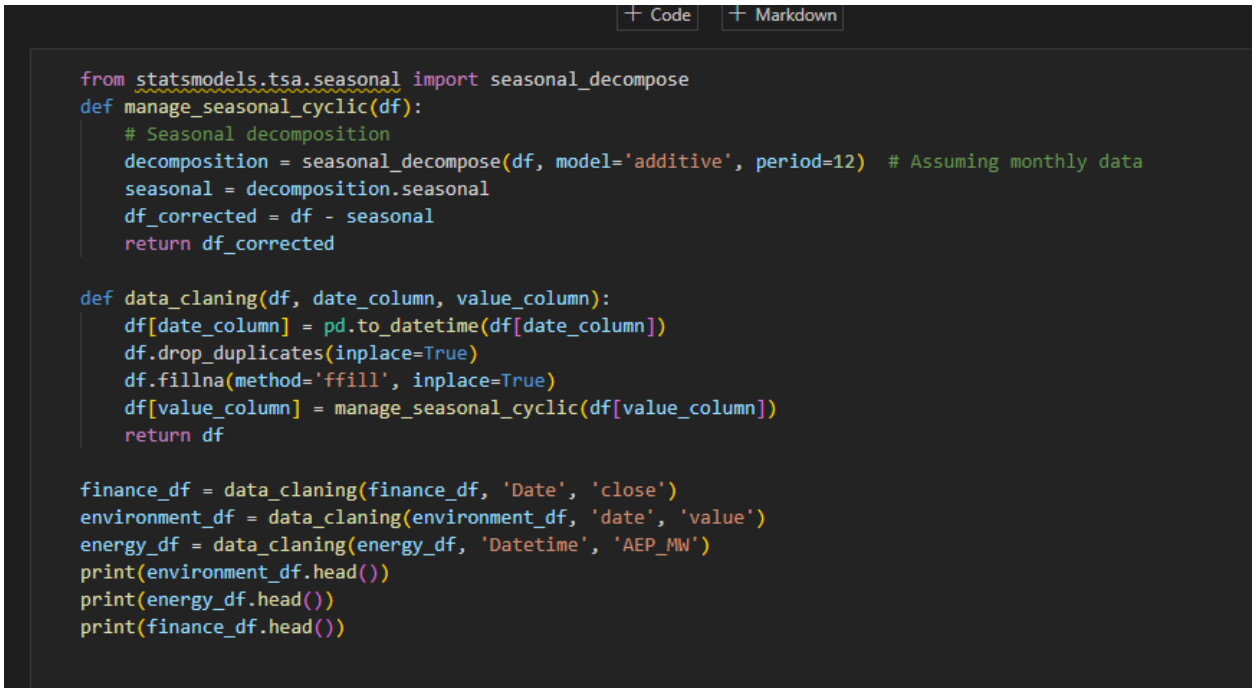
Manahil Kamran 21i-2668
Ali Arfa 21i-2669
DS-N

Project Report:

Comprehensive Forecasting System with User Interface for Multiple Sectors

Introduction:

The aim of this project was to train and analyze the performance of eight different forecasting models on three distinct types of datasets: finance, environment, and energy. These models included ARIMA, ANN, LSTM, SARIMA, SVR, Hybrid, ETS, and Prophet. The project also involved the development of a data visualization dashboard using HTML, CSS, JavaScript, and Bootstrap for presenting the results of model evaluation.



Model Training:

The training process involved significant computational resources, especially for models such as LSTM, Prophet, and Hybrid, which required over six hours to complete training due to their complexity. Other models typically completed training within three to four hours. The training

Manahil Kamran 21i-2668
Ali Arfa 21i-2669
DS-N
process required careful preprocessing and cleaning of the datasets to optimize model performance.

ARIMA

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.stattools import adfuller
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.arima.model import ARIMA

# Function to evaluate ARIMA model
def evaluate_arima_model(data, order):
    # Split data into train and test sets
    train_size = int(len(data) * 0.8)
    train, test = data[:train_size], data[train_size:]

    # Fit ARIMA model
    model = ARIMA(train, order=order)
    fitted_model = model.fit()
```

```python
ann(finance_df, 'Date', 'close', 'finance')
```

```
Epoch 1/50
C:\Users\aliar\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9_qbz5n2kfra8p0\LocalCache\lo
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
117/117 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - loss: 4376114.0000 - val_loss: 4625879.5000
Epoch 2/50
117/117 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 4610983.5000 - val_loss: 4320888.5000
Epoch 3/50
117/117 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 4093335.0000 - val_loss: 3523922.2500
Epoch 4/50
117/117 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 2980399.2500 - val_loss: 2324444.2500
Epoch 5/50
117/117 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 1857979.3750 - val_loss: 1232220.6250
Epoch 6/50
117/117 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 958222.1250 - val_loss: 649982.4375
Epoch 7/50
117/117 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 614382.6250 - val_loss: 490167.8438
Epoch 8/50
117/117 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 505423.0000 - val_loss: 456339.5000
Epoch 9/50
117/117 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 489345.5625 - val_loss: 441382.2500
```

Manahil Kamran 21i-2668
Ali Arfa 21i-2669
DS-N

Analysis and Performance Evaluation:

After training, the performance of each model was evaluated using metrics such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and loss. The analysis revealed varying levels of performance across different models and datasets. Some models performed better on certain datasets than others, highlighting the importance of selecting the appropriate model for each specific dataset.

```
Best ARIMA parameters: (2, 1, 2)
Best RMSE: 2613.0169685956607
(2, 1, 2)
2613.0169685956607
```

```
ann(environment_df, 'date', 'value', 'environment')
```

```
C:\Users\aliar\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
118/118 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 15.7408
Test Loss: 15.869555473327637
```

```
...
Epoch 50/50
117/117 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 334823.5625 - val_loss: 266317.3438
37/37 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 273090.7812
Test Loss: 285974.40625
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
+ Code + Markdown
```

Data Storage and Retrieval:

To facilitate easy storage and retrieval of model results, a SQLite database was utilized. The database schema was designed to store key information about each model, including dataset

Manahil Kamran 21i-2668
Ali Arfa 21i-2669
DS-N
type, model type, parameters, RMSE, loss, and image paths. This allowed for efficient retrieval of model results for display on the data visualization dashboard.

app > schema.sql

```sql
CREATE TABLE Category (
    id INTEGER PRIMARY KEY,
    name TEXT
);

CREATE TABLE Model (
    id INTEGER PRIMARY KEY,
    name TEXT,
    category_id INTEGER,
    rmse REAL,
    loss REAL,
    parameters TEXT,
    FOREIGN KEY (category_id) REFERENCES Category(id)
);

CREATE TABLE Attribute (
    id INTEGER PRIMARY KEY,
    name TEXT,
    value TEXT,
    model_id INTEGER,
    FOREIGN KEY (model_id) REFERENCES Model(id)
);

CREATE TABLE Image (
    id INTEGER PRIMARY KEY,
    path TEXT,
    model_id INTEGER,
    FOREIGN KEY (model_id) REFERENCES Model(id)
```

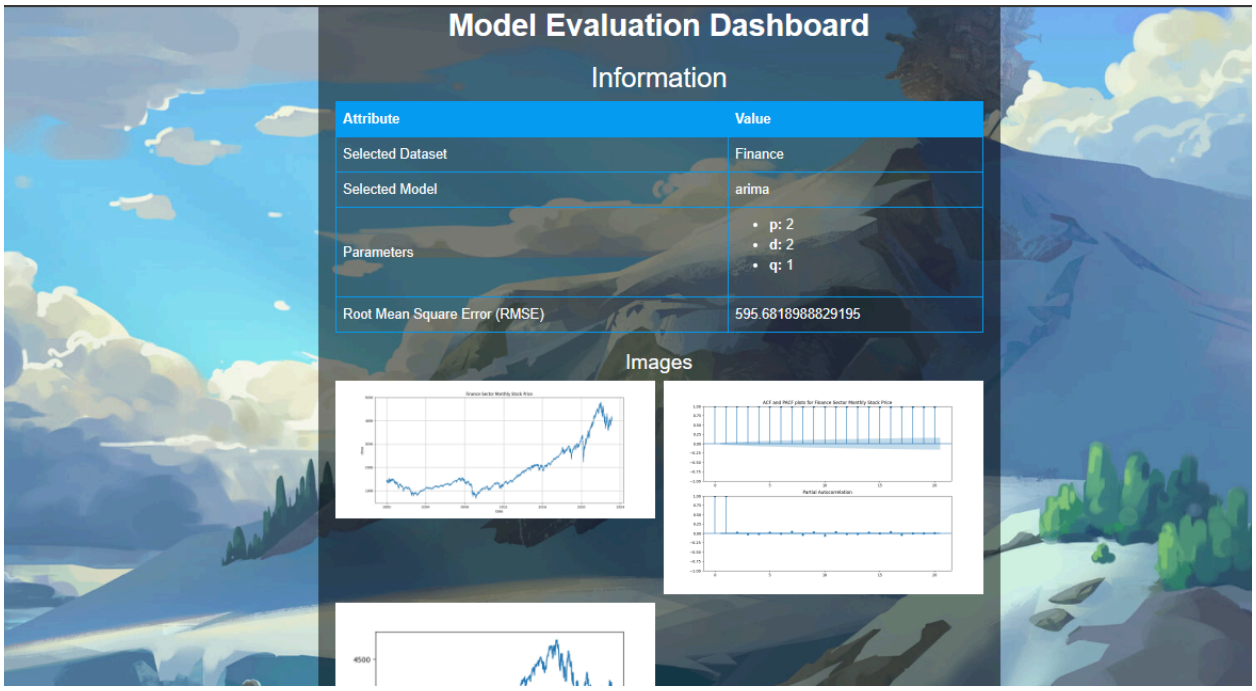
Manahil Kamran 21i-2668
Ali Arfa 21i-2669
DS-N

```
# Function to insert data from JSON into SQLite
def insert_data():
    with open('data2.json', 'r') as file:
        data = json.load(file)

    conn = sqlite3.connect('data.db')
    cursor = conn.cursor()

    for category_name, models in data.items():
        cursor.execute('INSERT INTO Category (name) VALUES (?)', (category_name,))
        category_id = cursor.lastrowid

        for model_name, attributes in models.items():
            rmse = attributes.get('rmse')
            loss = attributes.get('loss')
            parameters = json.dumps(attributes.get('best_params'))

            cursor.execute('''
                INSERT INTO Model (name, category_id, rmse, loss, parameters) VALUES
            ''', (model_name, category_id, rmse, loss, parameters))
```

Data Visualization Dashboard:

The data visualization dashboard served as a user-friendly interface for exploring the results of model evaluation. Users could select a dataset and model from dropdown menus, triggering the display of relevant performance metrics and graphs on the dashboard. The dashboard provided insights into the effectiveness of each model and allowed users to compare performance across different models and datasets.

Manahil Kamran 21i-2668
Ali Arfa 21i-2669
DS-N



Challenges and Solutions:

Preprocessing the datasets posed significant challenges due to various factors such as data inconsistency, missing values, and noise. Overcoming these hurdles required careful planning and implementation of preprocessing techniques. Below are some of the preprocessing challenges encountered and the solutions implemented to address them:

Manahil Kamran 21i-2668
Ali Arfa 21i-2669
DS-N

Data Cleaning: The datasets often contained missing values, outliers, and inconsistent formatting, which could adversely affect model training and performance. To address this, thorough data cleaning techniques such as imputation of missing values, outlier detection, and normalization were employed. For example, missing values were imputed using methods such as mean, median, or interpolation based on the nature of the data.

Feature Engineering: The raw datasets often contained a large number of features, some of which may not be relevant for forecasting. Feature engineering techniques were used to select or create relevant features that could improve model performance. This involved analyzing the correlation between features, transforming variables, and selecting the most informative features using techniques such as PCA (Principal Component Analysis) or feature importance scores.

Time Series Formatting: Since some datasets were time series data, special attention was required to ensure proper formatting for time series analysis. This involved converting timestamps to datetime objects, handling irregular time intervals, and aggregating data into appropriate time intervals for modeling purposes.

Model-specific Data Requirements: Certain models, such as LSTM and Prophet, have specific data requirements and preprocessing steps. For example, LSTM models require sequence data, so time series data was transformed into sequences with appropriate input and output sequences. Prophet models require a specific format with columns named 'ds' and 'y', representing the time series index and target variable, respectively. Adapting the datasets to meet these requirements required additional preprocessing steps.

Hurdles Faced:

Computational Resources: Training complex models like LSTM, Prophet, and Hybrid required significant computational resources and time. Limited computing power and long training times posed challenges in experimenting with different model architectures, hyperparameters, and preprocessing techniques. This extended the overall project timeline and required efficient resource management.

Optimization Challenges: Achieving optimal model performance, especially for models like LSTM and Prophet, was challenging. Tuning hyperparameters, selecting appropriate features, and minimizing loss functions required iterative experimentation and fine-tuning. This process was time-consuming and required a deep understanding of each model's architecture and parameters.

Model Evaluation: Evaluating model performance and comparing results across different models and datasets was complex. Determining the most suitable evaluation metrics,

interpreting results, and identifying the best-performing model required careful analysis and consideration of various factors such as dataset characteristics and modeling assumptions.

Dashboard Development: Developing a user-friendly data visualization dashboard involved integrating backend data processing with frontend design and interactivity. Ensuring seamless communication between the database, backend Python code, and frontend HTML/CSS/JavaScript components posed challenges in terms of data retrieval, formatting, and display.

Conclusion:

In conclusion, this project demonstrated the effectiveness of various forecasting models for analyzing finance, environment, and energy datasets. By training and evaluating multiple models and developing a user-friendly visualization dashboard, valuable insights were gained into the performance of each model and its suitability for different types of datasets. Moving forward, further optimization of model parameters and exploration of additional forecasting techniques could lead to improved performance and insights in future projects.